



The Holy Grail of Testing:
Have we found it?



Sérgio Freire

Solution Architect &
Testing Advocate

Xray



@darktelecom





**Quinta da Regaleira,
Sintra**

**Convent Of Christ,
Tomar**



Are we on the right quest?



Illusion 1

“Test everything”



In life, **and in testing**,
aiming for everything is not
happiness; it's a myth that
may lead to frustration.



The truth lake

Truth (infinite)

- **all possible uses of our product in all possible scenarios**
- **every possible bug**

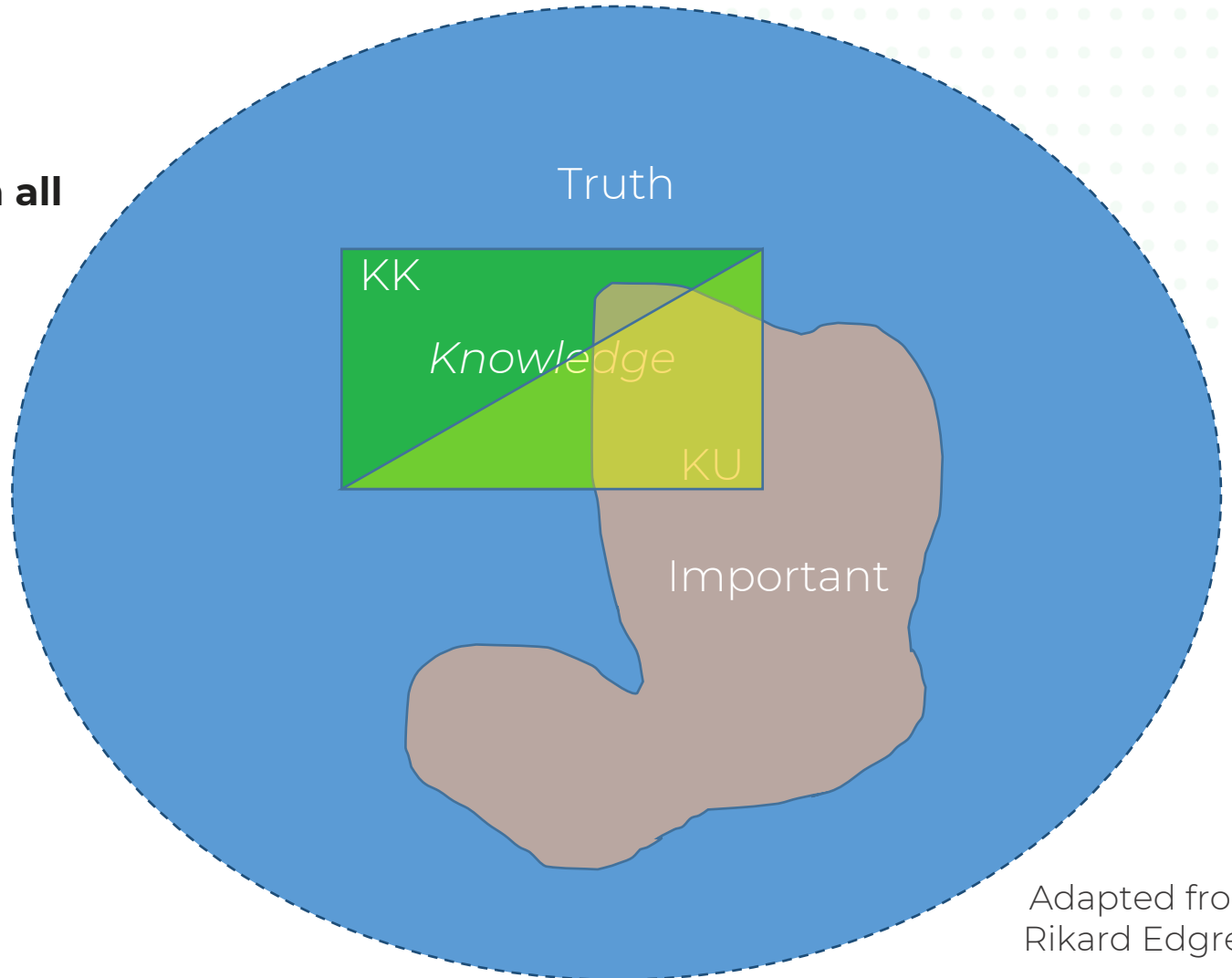
Knowledge

- KK+KU (known knowns, known unknowns)

Important

- risks we need to address

How to ensure we find many things around what's important?



Adapted from
Rikard Edgren



Illusion 2

“Automate all tests”

Illusion 3

“Bug free”
products



Image from Unsplash



@darktelecom

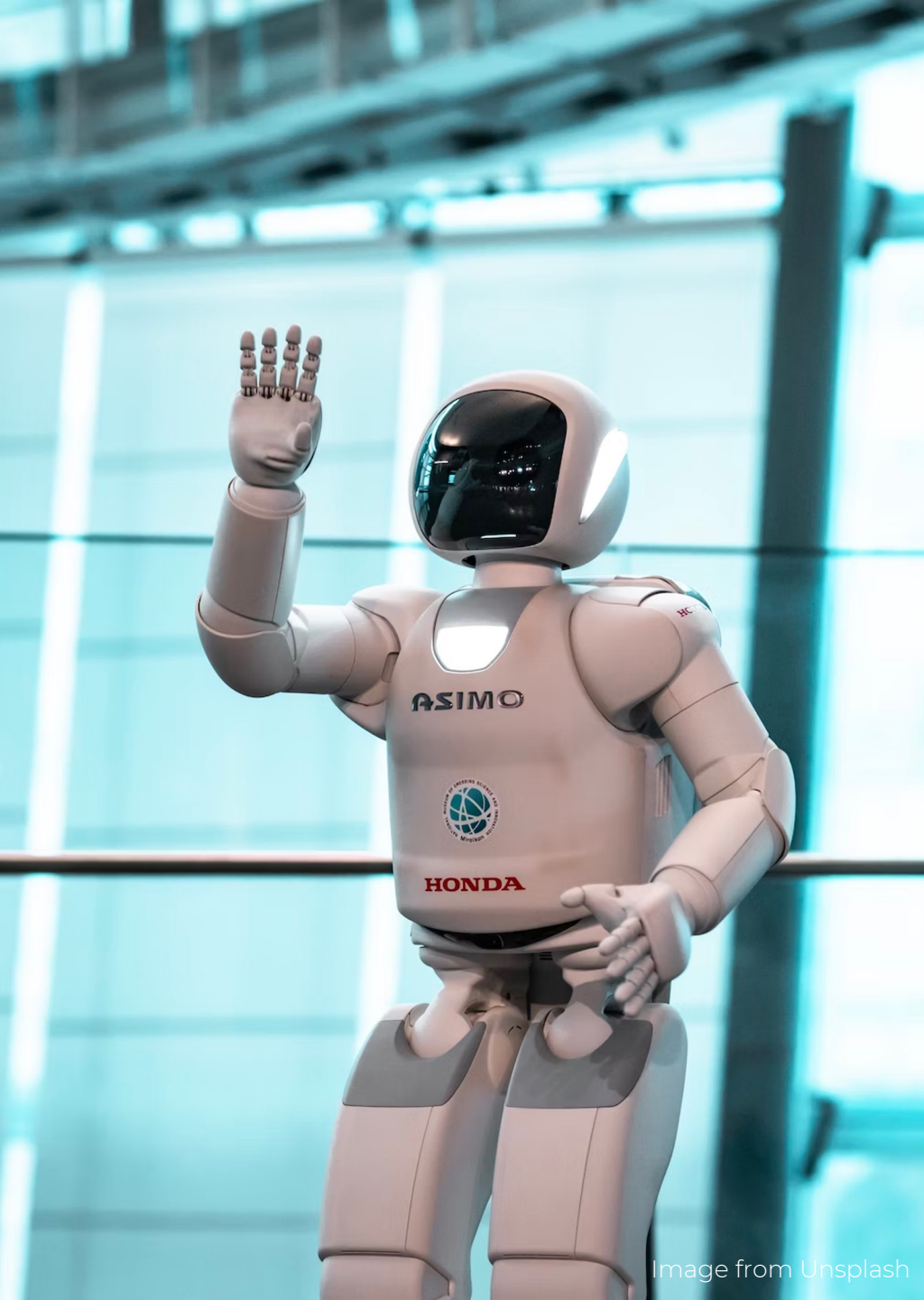


Image from Unsplash

Illusion 4

**“Tools will
replace us”**

Tools won't...

- Know your context
- Know your customers
- Know about the unimplemented features
- Change the work culture
- Improve the skills of your team
- ...

Are we addressing the right problem/need?

Who are the customers we aim to target and what do we know about them and their needs?

What are the timing, costs, and project constraints we have?

Enable team spirit, true collaboration, shared knowledge, incorporation of early feedback

Illusion 5

“Layers of abstraction and fancy UIs make it easier to test and less buggy”



Bugs increase with LoC (Lines of Code)

- 15-50 bugs per 1000 LoC (Steve McConnell)
- Cost of late fixes up to 30x higher

**Less code
doesn't mean
less bugs
though.**

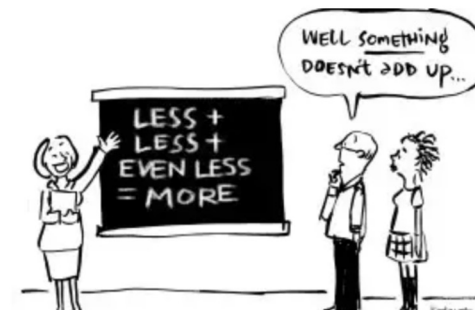


Jeff Nelson, Invented Chromebook, #Xoogler

Updated Jul 25, 2016 · Upvoted by Andreas Blixt, [programming since '96](#) and Greg Skinner, [25+ years of software engineering, mostly in C and Perl, mostly in the BSD networking stack and Unix utilities...](#)

Originally Answered: How many lines of code do professional programmers write per hour?

I recall a small project where I wrote about 1000 lines on the first day. On the second day, found a library that solved most of the problem for me, so I was able to cut the solution down to about 50 lines of code. On the third day, found a second library that solved virtually the exact problem, so it was cut down again to 2 lines of code. 2 lines of code in 3 days.



Source: [The Daily, 2SER | 5 April - from CPD](#)

I thought it was still a success, because I'd virtually eliminated the chance of any bugs and found a simple, elegant solution to the original problem. If I'd instead checked in the 1000 lines of code, it might have looked good on my code metrics, but I'd have probably spent another couple weeks debugging. @_jeff_nelson



**Have we
found the
holy grail of
testing?**

ChatGPT?





To test software, you can follow these steps:



1. Plan your testing approach: Decide on the type of testing you want to perform, create a test plan, and define your testing objectives and criteria.
2. Prepare your testing environment: Set up the testing environment with the required hardware, software, and network configurations.
3. Develop test cases: Create test cases that cover different scenarios, functionalities, and use cases.
4. Execute test cases: Run the test cases and record the results, including any defects or issues found during testing.
5. Report and track issues: Report any defects or issues found during testing and track their progress until they are resolved.
6. Retest fixed issues: Once issues have been fixed, retest them to confirm that the fixes were successful.
7. Evaluate testing results: Evaluate the testing results to determine if the software meets the requirements and is ready for release.

Depending on the type of software you want to test, you may need to use additional tools and techniques, such as automated testing tools, performance testing, security testing, and usability testing.

**Generic
answer...**

**What's
important
to me and
customers?**

Login Page

Please input your user name and password and click the login button.

User Name:

Password:

LOGIN



S

In the following HTML page what are the problems or potential bugs with it?

```
<html><head>
<title>Login Page</title>
<link href="demo.css" media="all" rel="stylesheet" type="text/css">
```



There are several potential issues with the provided HTML code. Here are some of them:

1. Accessibility issues: The form does not have any label elements associated with the input fields, which could cause accessibility problems for users with visual impairments who use screen readers to navigate the page. The form should use the "for" attribute in the label element to associate each label with its corresponding input field using the "id" attribute.
2. Security issues: The login function is executed on the client-side, which means that the username and password are transmitted in plain text over the network. This is a security risk, as it can allow attackers to intercept and read the sensitive information. To prevent this, the login function should be executed on the server-side using a secure communication protocol, such as HTTPS.
3. Validation issues: The login function does not perform any input validation on the username and password fields, which could lead to issues such as SQL injection attacks, cross-site scripting attacks, and other security vulnerabilities. The login function should validate and sanitize the user input before processing it.
4. Usability issues: The form does not provide any feedback when the login attempt fails. This can make it difficult for users to know how to correct the issue. The form should provide feedback when the login attempt fails.
5. HTML syntax issues: The code does not have any closing tags for the head, body, or html elements. It is recommended to use proper HTML5 syntax, such as <input type="text"> for input elements to indicate the input type.

Missed biggest issue: hardcoded credentials!

```
<script type="text/javascript">
  function login(username, password) {
    if (username == "demo" && password == "mode") {
      window.location = "welcome.html";
    } else {
      window.location = "error.html";
    }
  }
</script>
```



Other things missed

- What is the purpose of this page?
- What permissions do we need to have in place? Do we need roles?
- What are the authentication mechanisms supported (e.g., login/pass, oauth, Google/FB login)
- What should happens when it succeeds? And when it fails?
- What's the process to recover a password?
- Is there a signup process? Is there a backoffice somewhere?
-

What can we do then?

Leverage human testing with tools

Tools to...

- Provide some **insights** about the code and help write better code, from the start
- Test more **combinations**
- Perform tests/checks in **parallel**
- Address **types of tests** that otherwise would be impossible to test (e.g., load testing)
- Depict strange **patterns** (e.g., visual testing)
- ...

The holy grail of testing
comes from **humans**
leveraged by **tools** working
in a true **team**
environment.





Thank you!

Find us at

getxray.app | mail@getxray.app | [@XrayApp](https://twitter.com/XrayApp)



[@darktelecom](https://twitter.com/darktelecom)

Blog: sergiofreire.com

